National Institute of Technology, Rourkela

Department of Electronics and Communication Engineering

May-2013

SOLAR TRACKER SYSTEM FOR PV MODULE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Bachelor of Technology in Electronics and Instrumentation Engineering

Submitted by,

Pamarthi Kranthi Kumar

(109EI0083)

Ritika Ojha

(109EI0566)

Under the guidance of,

Dr. Umesh C. Pati

National Institute of Technology, Rourkela

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the Thesis Report entitled SOLAR TRACKER SYSTEM FOR PV MODULE submitted by Pamarthi Kranthi Kumar (109EI0083) and Ritika Ojha (109EI0566) of Electronics and Instrumentation Engineering during May 2013 at National Institute of Technology, Rourkela is an authentic work performed by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Dr. Umesh C. Pati

Dept. of Electronics and Communication Engineering

National Institute of Technology, Rourkela

National Institute of Technology, Rourkela

Department of Electronics and Communication Engineering

ACKNOWLEDGEMENT

Firstly, we wish to express our deep sense of gratitude to Dr. Umesh C. Pati, Dept. of Electronics and Communication Engineering, National Institute of Technology, Rourkela, our guide, for his consistent encouragement, incalculable guidance and cooperation to carry out this project, and for giving us an opportunity to work on this project and providing us with a great environment to carry our work in ease.

We would also like to thank our friends, for providing insightful comments and helping enhance the quality of thesis, and family, for being understanding and supportive throughout. Their help cannot be penned with words.

Pamarthi Kranthi Kumar (109EI0083)

Ritika Ojha (109EI0566)

Dept. of Electronics and Communication Engineering

National Institute of Technology, Rourkela

ABSTRACT

The project proposes FPGA implementation of a novel approach to track maximum power point of solar photo-voltaic array and a novel approach to track the solar position and find a suitable position to attain more energy in comparison with those in a fixed position. The approach uses Kalman Filter algorithm to track maximum power point, motor position and piston position. The finite state machine is Melay type and includes five states. Using the proposed technique, the MPPT can be tracked up to the efficiency of 97% within a time of about 4.5ms. The position of PV array is tracked to an error of ±2%mostly. Experiments have been carried out in partially shaded, falling irradiance level conditions. The proposed method is simple and cost effective in comparison to systems using GPS to track the position.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter-1

INTRODUCTION

1.1 Introduction

1.2 Literature Survey

1.3 Motivation

1.4 Objective

1.1 Introduction

Renewable energy sources are more abundant in comparison to the traditional fossil fuels and theoretically they can easily supply the world's energy needs. Earth's surface receives 89PW of total incoming solar insolation. It is not possible to harness all of it, but even if we capture a part of it, even less than 0.02%, would be enough to meet our requirements. The solar radiations received can be converted into electrical energy by using Photo-Voltaic Cells or Solar Cells which generate electrical energy by Photo-Voltaic effect i.e. building of Voltage or Direct Electric Current in a semi-conductor material when exposed to light. Photovoltaic power generation is achieved using solar panels which are composed of a number of solar cells made up of a photovoltaic material. Materials currently used for photovoltaic include amorphous silicon, copper indium gallium selenide/sulphide, cadmium telluride, monocrystalline silicon, and polycrystalline silicon. Due to the continuous increase in demand for renewable energy sources, the manufacturing of Solar Cells and Photo-Voltaic Arrays has considerably advanced over the past few years.

The output of a Photo-Voltaic panel is non-linear in nature as well as the output greatly varies with environmental changes like temperature, solar insolation, etc. This poses a huge threat while using it. The entire Photo-voltaic array does not receive equal amount of radiations at all times. Sometimes parts of the array is under shading due to clouds, tree, towers, dust, etc. This results in occurrence of multiple peaks in the Power v/s Voltage characteristics of the array which hinders the proper functioning on maximum power point tracker. Considerable power loss will be experienced if local maximum power point is tracked and nor the global maximum. Hence it is pertinent to track the optimal operating voltage of the Photo-Voltaic array for better efficiency of PV generators.

The PV array gives different output at different time of day for different orientations depending upon the amount of sunlight falling on the module and on the angle at which rays fall on it. Thus the angle at which the module is placed with respect to the ground can be optimized so as to obtain maximum output from the module at all time.

1.2 Literature Survey

Utilization of energy obtained from solar panel has been an important topic of research in recent times because of the growing requirement of sustainable resources of energy. The output of solar panel is very low hence optimization is required. This is achieved by using maximum power point tracking and adjusting of panels at an orientation which will yield maximum power output.

Shubhajit Roy Chowdhury, Dipankar Mukherjee, Hiranmay Saha, in their paper "FPGA based maximum power point tracker of partially shaded solar photovoltaic arrays using modified adaptive perceptive particle swarm optimization" [1], have discussed several optimization techniques for maximum power point tracking. Their tracking logic is based on a modified version of APPSO (Adaptive Perceptive Particle Swarm Optimization) technique. This was implemented on a FPGA board.

They studied various MPP tracking techniques: Hill-climbing technique, Perturb and Observe technique and Incremental Conductance. Hill-climbing technique is done by using a perturbation in the duty ratio of power converter. The Perturb and Observe technique is done by using a perturbation in the operating voltage of the photovoltaic array. Incremental conductance is another technique based on the fact that slope of photovoltaic array power curve tends to be zero at the maximum power point; the slope is positive on the left of the MPP and negative on the right.

Trishan Esram and Patrick L. Chapman, in their paper "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques" [6] and Roberto Faranda and Sonia Leva, in their paper "Comparison of MPPT techniques for PV systems" [7], have provided comparisons of a number of power point tracking techniques and worked on efficiency of each.

R. E. Kalman, in his paper "A New Approach to Linear Filtering and Prediction Problems" [2], described an effective computational (recursive) solution to the discrete-data linear filtering problem by estimating that state of a process that minimizes the mean squared error. These filters are extensively used in autonomous and assisted navigation. Greg Welch and Gary Bishop, in their paper "An Introduction to the Kalman Filter" [3], provided an applied demonstration of discrete Kalman Filters.

1.3 Motivation

The need for efficient renewable sources of energy has been increasing every day. This has prompted the researched to work for increasing developing better solar cell materials and improving the efficiency of the panel by implementing better control mechanisms. The challenge of new area and its requirement in society was the the motivation behind the project.

1.4 Objective

The main objective of the project is to achieve maximum possible output from the solar panel at all times of day. This is achieved in three steps. First multiple power point tracking technique would be studied and implemented. Then the panel will be rotated, horizontally as well as vertically, to function in a proper orientation. The above would be achieved by implementing Kalman Filters.

Chapter-2

PV MODULE

2.1 Design of PV array

2.2 Characteristics of PV array

2.1 Design of PV array

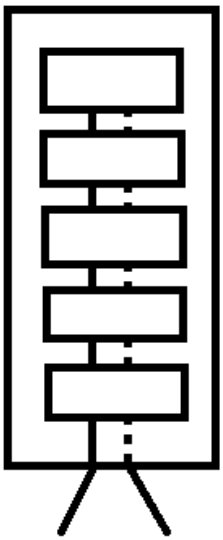
A single cell produces only a voltage of 0.5-0.6V and a few watts of power, hence it is of little use. To produce a larger voltage, a number pre wired cells in series all encased in tough weather resistant package to form a module. When photo-voltaics are wired in a serie, they all carry same current and their voltages addd. The overall voltage of module V_{module} is formed by

$$V_{module}=n(V_d-IR_s) \qquad \text{n:no of cells}$$

The PV module supplied to us contains 5 cells in series. We have 2 such panels

Since the module is fabricated hence it is not possible to collect individual cell data. To overcome this we can use the given PV module and obtain the readings by shading all cells except cells of interest and collect data. But it has certain disadvantages

- Difficult to carry out
- Readings not very reliable
- Hectic while taking data
- Original panel does not come with gates to cover and adding gates will reduce production



$(V_{out})_{max}=2.75V$

Fig 2.1 Solar Panel

2.2 Characteristics of PV array

PV array consists of collection of numerous solar cells in series or parallel to get the desired voltage and current. Following figure shows the equivalent circuit model of a solar cell. Rsh is very large, Rs is very small and both can be ignored in order to simplify the electrical model.

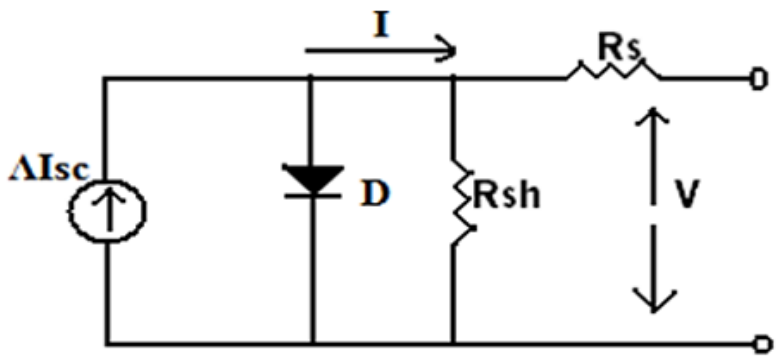


Fig 2.2 Solar cell equivalent circuit

The simplified equation [5] to describe the PV panel is given as

$$I = I_{SC}\left\{\lambda - \frac{1}{\exp\left(\frac{qA}{kT}\right)}\left(\exp\left(\frac{qAV}{kTV_{OC}}\right) - 1\right)\right\}$$

Where, Voc and Isc are open circuit voltage and current values at 1kW/m2 and 250C. T is the temperature of array in 0 C, q is the elementary charge, k is the Boltzmann constant, λ is irradiance in kW/m2 and A is a constant, generally taken as 0.2464. V and I are the array output voltage and current.

A general I-V curve is shown in the following figure under given conditions i.e. irradiance of 1kW/m2 and temperature of 250C there is one point on the I-V curve which gives Maximum Power Point because it maximizes the area under the curve. A general P-V curve is also shown. The PV panel considered has Voc =

22 V and Isc = 1.3 A at 1kW/m2 and 250C.

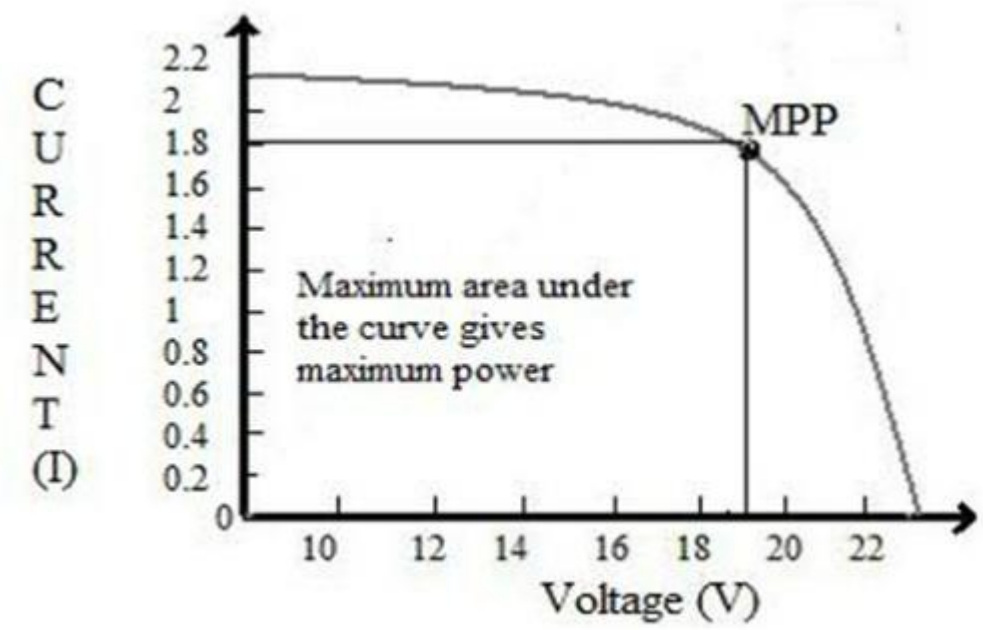


Fig 2.3 Current vs Voltage characteristics of panel

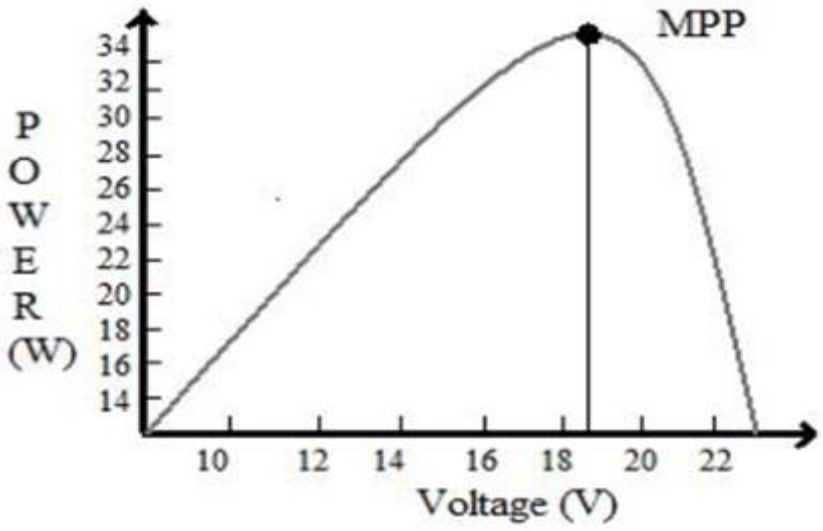


Fig 2.4 Power vs Voltage characteristics of panel

Chapter – 3

KALMAN FILTERS

3.1 Introduction to Kalman Filters

3.2 Kalman Filters as MPPT algorithm

3.3 Implementation of Kalman Filters in FPGA

3.1 Introduction to Kalman Filters

Kalman filter provides stochastic estimation in noisy environment. The kalman filter operates on estimating states by using recursive time updates & measurement updates over time. Noise effect in the system is decreased due to recursive cycles which finally lead to the true value of measurement. Following figure shows the generic block diagram of Kalman Filter.

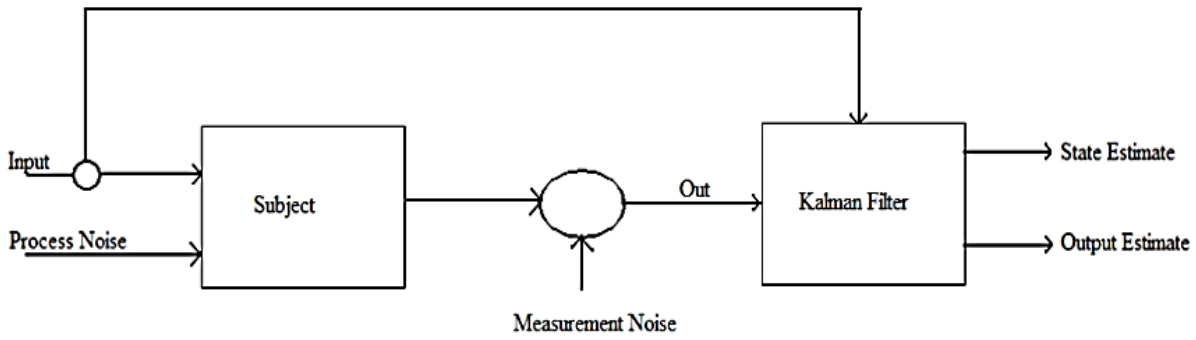


Fig 3.1 Generic block diagram to describe Kalman Filter algorithm

Kalman Gain:-

Kalman Gain is the blending factor that minimizes the posteriori error covariance. The gain is devised based on 2 factors

1) If measurement error covariance approaches zero, it should weigh heavily

2) If posteriori error covariance approaches zero it should weigh less heavily

3.2 Kalman Filters as MPPT Algorithm

According to PV curve of PV cell, power increases with a gradual positive slope until it reaches one optimal point and then decreases after that steeply. We pass voltage from module into Kalman Filter.

Let V_{act} be the process, V_{act}^{t} be the known state. Then

$$V_{act}^{t+1}=V_{act}^{t}+M(\Delta P/\Delta V)+w$$

Here A=1; B=M

Let V_{ref} be module output

$$V_{ref}^{t+1}=V_{act}^{t+1}+v$$

Voltage estimate priori at t+1 is

$$V_{act}^{t+1}=V_{act}^{-t}+M(\Delta P^{-t}/\Delta V^{-t}) \quad (1)$$

The process noise is assumed to be zero initially

Priori estimate of error covariance

$$z_{t+1}^{-}=z_t+Q \quad (2)$$

Measured voltage is

$$V_{ref}^{t+1}=V_{act}^{-t+1}+v \quad (3)$$

Here C=1

Equation (1) and (2) form the prediction state.

The Kalman gain is

$$K_{t+1}=z_t^{-}c^T/cz_t^{-}c^T+R$$

$$K_{t+1}=z_{t+1}^{-}/z_{t+1}^{-}+R$$

$$K_{t+1}=z_{t+1}^{-}(z_{t+1}^{-}+R)^{-1} \quad (4)$$

Posteriori voltage estimate is

$$V_{act}^{t+1}=V_{act}^{-t+1}+K_{t+1}[V_{ref}^{t+1}-V_{act}^{-t+1}] \quad (5)$$

Posteriori error covariance estimate is

$$Z_{t+1}=Z_{t+1}^{-}(1-K_{t+1}) \quad (6)$$

Equations (4), (5) and (6) will form the correction states.

3.3 Implementation of Kalman Filters in FPGA

Judging from the equations, we would need single precision adders, subtractors, multipliers and inverse.

Since for solar PV array the quantities as all single matrices of orders 1X1

Construction of basic building blocks:

I) Single precision float adder

Aim: To design a single precision float adder using VHDL

Equipment required: Modelsim 6.6d starter edition

Requirement: Two single precision float point numbers should be added which means input width is 32 bits each and output width is 32 bits

Inputs: clock=1 bit

Data a=32 bits

Data b=32 bits

Output: result=32bits

Observation: Data a=1

Data b=1

Result=2

Conclusion: The adder for single precision float numbers has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed adder showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol)

II) Single precision float subtractor

Aim: To design a single precision float subtractor using VHDL

Equipment required: Modelsim 6.6d starter edition

Requirement: Two single precision float point numbers should be subtracted which means input width is 32 bits each and output width is 32 bits

Inputs: clock=1 bit

Data a=32 bits

Data b=32 bits

Output: result=32bits

Observation: Data a=2

Data b=2

Result=0

Conclusion: The subtractor for single precision float numbers has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed subtractor showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol)

III) Single precision float multiplier

Aim: To design a single precision float multiplier using VHDL

Software required: Modelsim 6.6d starter edition

Requirement: Two single precision float point numbers should be multiplied which means input width is 32 bits each and output width is 32 bits

Inputs: clock=1 bit

Data a=32 bits

Data b=32 bits

Output: result=32bits

Observation: Data a=2

Data b=1

Result=2

Conclusion: The multiplier for single precision float numbers has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed multiplier showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol)

IV) Single precision float added/subtractor

Aim: To design a single precision float adder and subtractor using VHDL

Software Required: Modelsim 6.6d starter edition

Requirement: Two single precision float point numbers whose width are 32 bits each and output width is 32 bits. Based on a particular signal bit the adder and subtractor should function. When the bit is '1', it should add and when the bit is '0', it should subtract.

Inputs: clock=1 bit

Add_sub=1 bit

Data a=32 bits

Data b=32 bits

Output: result=32bits

Observation:

i. When add_sub=1

Data a=2

Data b=3

Result=5

ii. When add_sub=0

Data a=2

Data b=3

Result=-1

Conclusion: The adder and subtractor for single precision float numbers has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed adder/subtractor showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol)

V) 2:1 Multiplexer

Aim: To design a 2:1 Multiplexer using VHDL

Software used: Modelsim 6.6d starter edition

Requirement: The MUX should take two 32 bit inputs and with the help of a select line, it should output the desired input, giving input1 when signal line is '0' and input2 when signal line is '1'

Inputs: clock=1bit

Select=1bit

Data a=32 bits

Data b=32bits

Output: Result=32bits

Observation:

i)When Select=’0’

Data a=2

Data b=3

Result=2

ii)When Select=’1’

Data a=2

Data b=3

Result=3

Conclusion: The 2:1 multiplexer has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed multiplexer showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol)

VI) 1:1 Demultiplexer with different input and output widths

Aim: To design a 1:1 Demultiplexer with different input and output widths using VHDL

Software used: Modelsim 6.6d starter edition

Requirement: The DEMUX takes a 32 bit input and with the help of two select lines and four switch lines, it gives 8 bit output, breaking the 32 bit input into four 8 bit outputs through a single line.

Inputs: clock=1bit

Select=2bit(s0,s1)

Switch=4bits(z0,z1,z2,z3)

Data=32 bits

Output: Result=8bits

Observation:

i)When Select=”00”

z1=’0’

Result=Data(31 down to 24)

ii)When Select=”01”

z1=’0’

Result=Data(23 down to 16)

iii)When Select=”10”

z1=’0’

Result=Data(15 down to 8)

iv)When Select=”11”

z1=’0’

Result=Data(7 down to 0)

Conclusion: The 1:1 demultiplexer has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed multiplexer showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol).

VII) Register to hold data

Aim: To design a register to hold data of 32 bits in VHDL

Software required: Modelsim 6.6d starter edition

Requirement: A register has to be designed such that it holds the data until an iteration is complete. The register needs clock, a reset line to reset back to register and a load line which loads the value of input into register only when it is high.

Inputs: Clock=1bit

Reset=1bit

Load=1bit

Data=32bits

Output: Result=32bits

Observation:

i)When Reset='1'

Data=X

Result=0

ii)When Reset='0'

Load='0'

Data=X

Result=0

iii)When Reset='0'

Load=’1’

Data=2

Result=2

Conclusion: A 32bit register has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. The designed register showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol).

VIII) Register to hold data and send out enable signal

Aim: To design a register to hold data of 32 bits and send out an enable signal in VHDL

Software required: Modelsim 6.6d starter edition

Requirement: A register has to be designed such that it holds the data until an iteration is complete. The register needs clock, a reset line to reset back to register and a load line which loads the value of input into register only when it is high.

Inputs: Clock=1bit

Reset=1bit

Load=1bit

Data=32bits

Output: Result=32bits

Enable=1bit

Observation:

i)When Reset=’1’

Data=X

Result=0

Enable=0

ii)When Reset=’0’

Load=’0’

Data=X

Result=0

Enable=0

iii)When Reset=’0’

Load=’1’

Data=2

Result=2

Enable=1

Conclusion: A 32bit register has been designed and simulated in modelsim 6.6d and the test-bench waveforms recorded. It send out a signal as soon as data is stored in it The designed register showed the desired characteristics. The decimal to binary conversions are verified using a float converter (IEEE 754 protocol).

IX) Comparator

Aim: To design a comparator using VHDL

Software required: Modelsim 6.6d starter edition

Requirement: To design a comparator which gives a two bit signal by comparing the two 32 bit input signals

Inputs: Clock=1bit

Reset=1bit

Data a=32bits

Data b=32bits

Output: Comp=2nits

Obseravations:

i)When Reset=’1’

Data a=X

Data b=X

Comp=”00”

ii)When Reset=’0’

Data a<Data b

Comp=”01”

iii)When Reset=’0’

Data a>Data b

Comp=”10”

ii)When Reset=’0’

Data a=Data b

Comp=”11”

Design of Kalman filter:

Equipment Required: Modelsim 6.6d starter edition and quartus II

Taking the equations into consideration, the filter is designed in 5 stages.

Stage-1:Priori Voltage Estimate at t+1

$$V_{act}^{-t+1}=V_{act}^{-t}+MU(t)$$

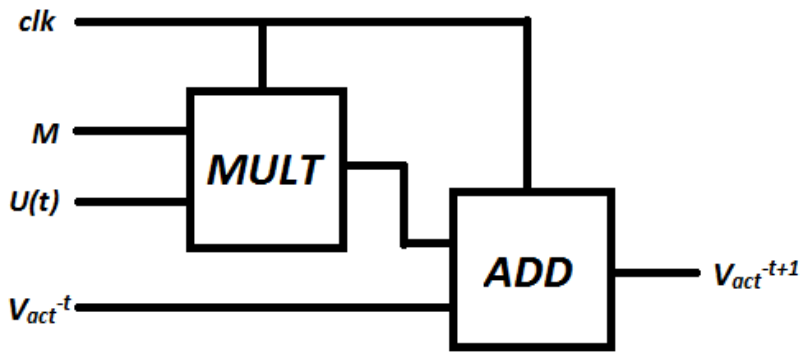


Fig 3.2 Stage-1 of Kalman Filter Design

Stage-2:Priori Error Covariance Estimate at t+1

$$z_{t+1}^{-}=z_t+Q$$

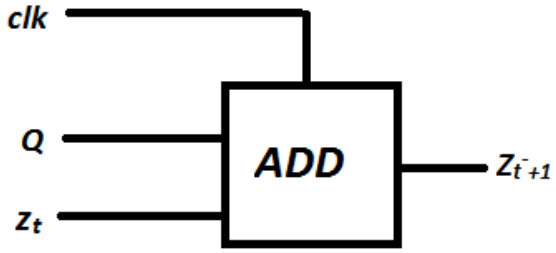


Fig 3.3 Stage-2 of Kalman Filter Design

Stage-3:Kalman Gain

$$K_{t+1}=z_{t+1}^{-}(z_{t+1}^{-}+R)^{-1}$$

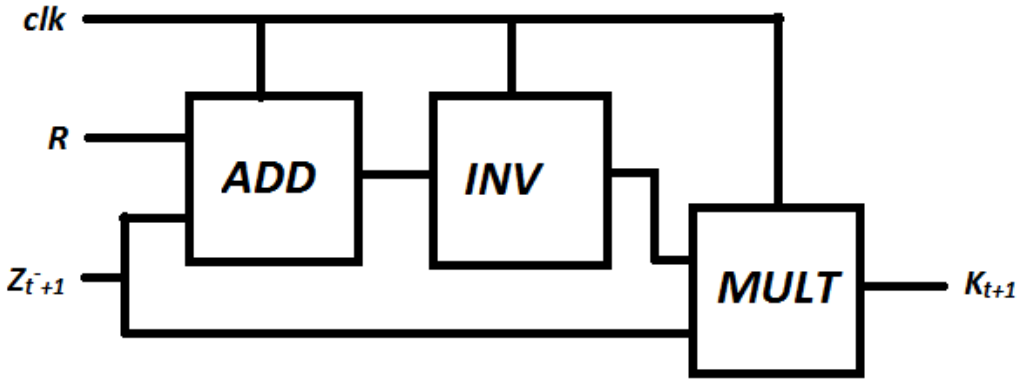


Fig 3.4 Stage-3 of Kalman Filter Design

Stage-4:Poteriori Voltage Estimate at t+1

$$V_{act}^{t+1}=V_{act}^{-t+1}+K_{t+1}[V_{ref}^{t+1}-V_{act}^{-t+1}]$$

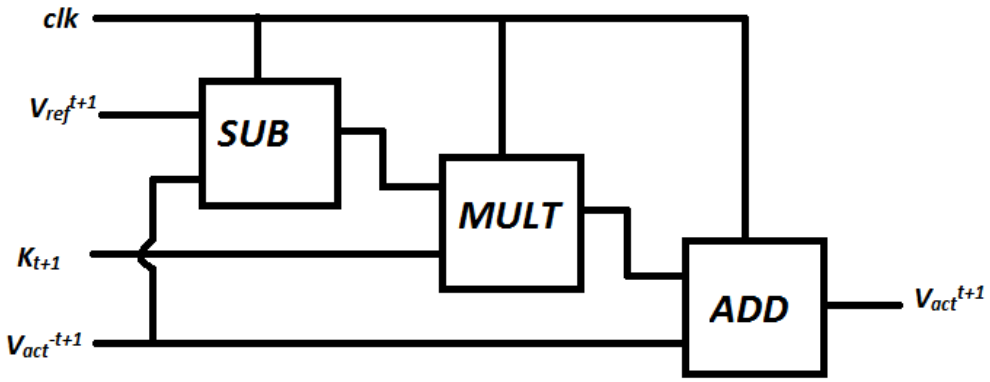


Fig 3.5 Stage-4 of Kalman Filter Design

Stage-5:Poteriori Error Covariance Estimate at t+1

$$z_{t+1}=z_{t}^{-}{}_{+1}(1-K_{t+1})$$

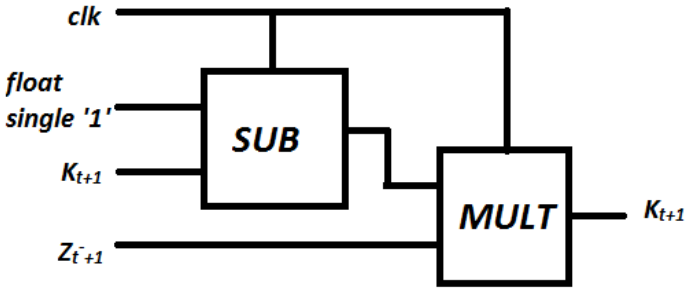
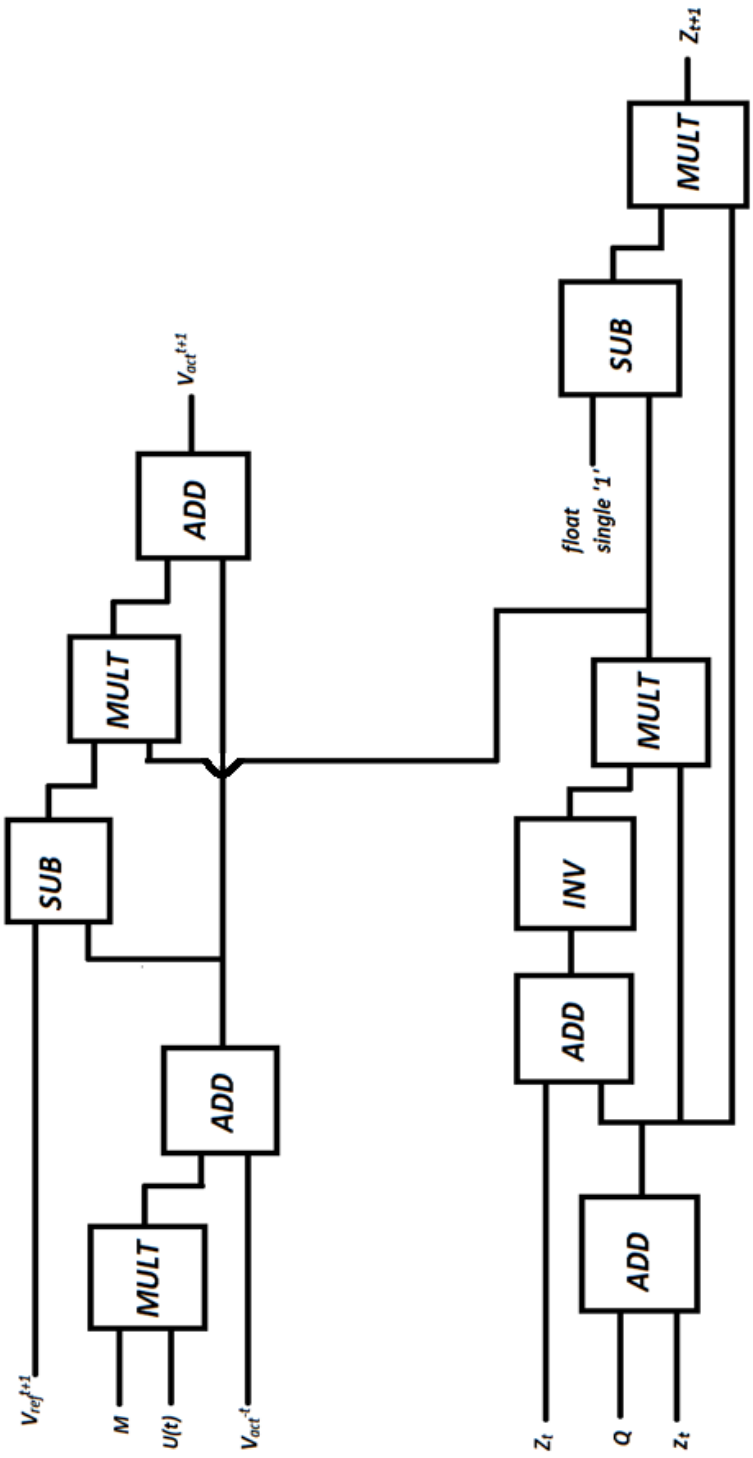


Fig 3.6 Stage-5 of Kalman Filter Design



*clock is supplied to every block

Fig 3.7 Kalman Filter Design

Conclusion: The designs have been compiled using Quartus-II and tested using panel output data.

Chapter – 4

ROTATION MECHANISM FOR PV MODULE

4.1 Design of Mount

4.2 Self-regulatory control for mount

4.3 Motor Algorithm

4.4 Kalman Filter for motor algorithm

4.5 FPGA implementation of motor control

4.6 Cylinder Algorithm

4.7 Kalman Filter for cylinder algorithm

4.8 FPGA implementation of cylinder control

4.1 Design of Mount

We require a suitable mount which could support rotation of the module towards sun. This can be achieved by using a motor and solenoid powered cylinder. The piston length of cylinder will be calculated during the implementation for a rectangular module,

1) Support of module on a shaft attached to a motor to rotate the module in horizontal direction. The contact of module and shaft is made in such a way that the module is hinged in the shaft to facilitate its vertical rotation.

2) A solenoid powered cylinder is mounted in the shaft such that the piston head is attached exactly at a quarter length of the module from hinge centre. This helps for vertical movement of module as we do not require entire rotation. A rotation of about 30^{o} odd is required as we don't want the module to be perfectly perpendicular to the incident light.

For maximum output power, we use both these steps to position PV module.

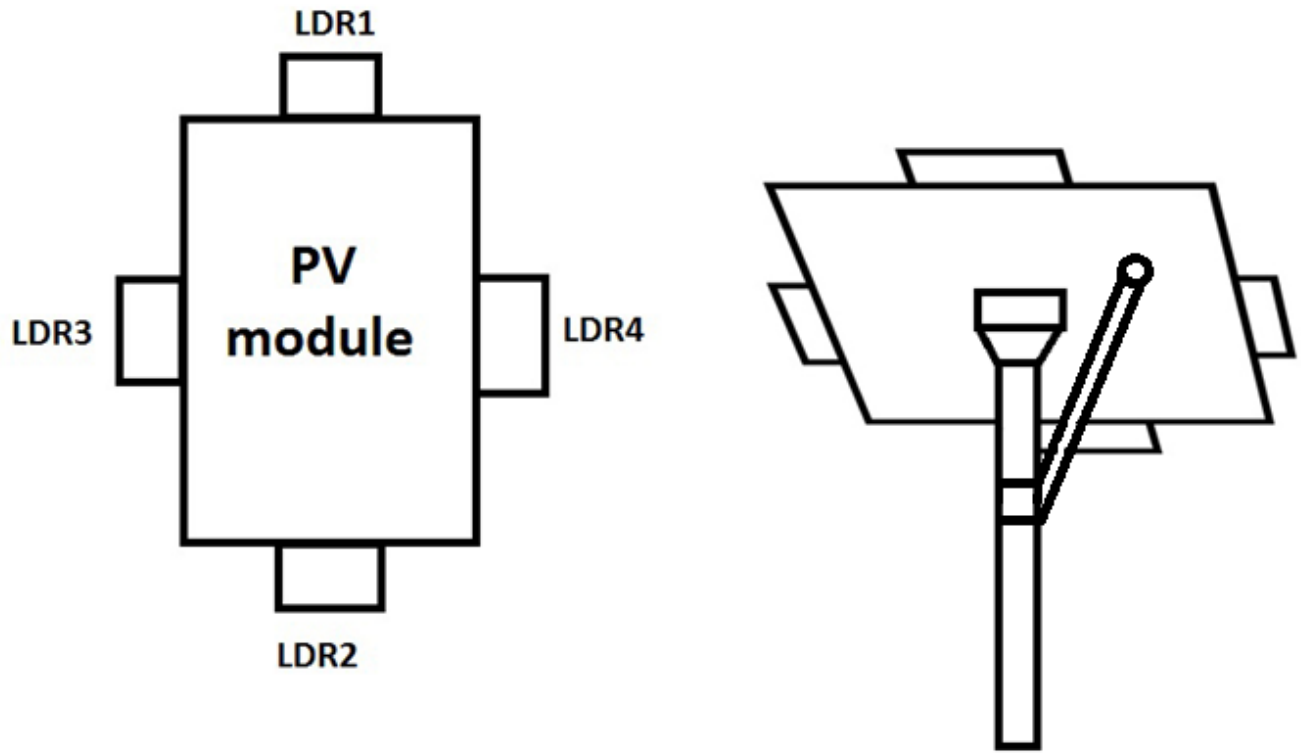


Fig 4.1 Design of mount (a) Top View (b) Rear View

From the figure we can observe how the shaft is hinged to the module and how the solenoid powered cylinder piston head is attached to the module. The piston will be completely out in the noon hours and completely in the early hours of sunrise and sunset.

For technical feasibility, we consider only 4 LDR circuitry in the module. The output of LDR3 and LDR4 is used for horizontal displacement and LDR1 and LDR2 is used for vertical displacement.

Let V1 and V2 be voltages of LDR 1 and LDR2 and V3 and V4 be voltages of LDR3 and LDR4.

4.2 Self-regulatory control for mount

Two essential components used for the control of the mount in respect to position are motor and the cylinder.

Uses of motor:-

- Motor is used to rotate the panel to required position
- Motor should be capable of rotating the panel both in CW and ACW direction.
- The direction is found out based in LDR sensor output
- Self-regulatory control is to be achieved using Kalman filter to estimate the required rotation.

Uses of cylinder:-

- Cylinder is used to elevate the panel to the required position.
- Cylinder should be capable of moving the panel in both extending and retracting direction
- The direction is found out based on LDR sensor output.
- Self-regulatory control is to be achieved by using Kalman filter to estimate required rotation

4.3 Motor Algorithm

This is the phase-1 of movement control. LDR sensor 3 and 4 are used as primary control input to the system. Voltages at hand are V3 and V4. To find necessary rotation, first we need to find out which voltage is more. The rotation takes place towards higher voltage side.

The magnitude of rotation is known from the difference of average of 2 known voltages and difference of 2 known voltages.

Ideally the rotation stops when average value is equal to low voltage.

Let V_{m1}= V3-V4

V_{m2}= (V3+V4)/2

When V_{m1}>0, V3>V4, rotation takes place towards LDR3 an stops when V4=Vm2

When V_{m1}<0, V3<V4, rotation takes place towards LDR4 and stops when V3=Vm2

4.4 Kalman filters of motor algorithm

Let Θ be the process and Θ_{t-1} be the known state

Measured variable is the high irradiance position at a particular time during the day.

Let it be Θ_{ref}

We have,

$$\Theta_t = \Theta_{t-1} + MU$$

Here A=J and B=M

Rotation estimate priori at ‘t’ is,

$$\hat{\Theta}_t^- = \hat{\Theta}_{t-1} + MU \qquad (1)$$

The process noise is assumed to be 0 initially

Priori estimate of error covariance is

$$Z_t^- = Z_{t-1} + Q \qquad (2)$$

Measured rotation is

$$\Theta_{reft} = \Theta_t^- + MU \qquad (3)$$

Here c=1

Equation (1) and (2) form prediction state

The Kalman Gain is calculated from formulae

$$K_t = z_t^- c^T / (c z_t^- c^T + R)$$

Since c=I in our requirement

$$K_t = z_t^- / (z_t^- + R)$$

$$K_t = z_t^- (z_t^- + R)^{-1} \qquad (4)$$

Posteriori rotation estimate is

$$\Theta_t = \Theta_t^- + K_t[\Theta_{reft} - \Theta_t^-] \qquad (5)$$

Equation (4), (5) and (6) form correction step

Control Element (MU)

From equation (1) in previous algorithm

$$\Theta_t = \Theta_{t-1} + MU$$

MU is dimensionless quantity in radians.

Sources in hand are two voltages V_3 and V_4

From control algorithm, control factor

$$U=V_{m2}-(V_3-V_4)$$

The unit of U is thus Volts(V)

The unit of M should be degrees/V

Ideal case: Let V_3 has maximum and V_4 has minimum measurable voltage on the panel

$$(V_{max}-V_{min})V=180$$

$$1V=180/(V_{max}-V_{min})$$

Number of degrees/Volt=$1/(V_{max}-V_{min})/180$

Number of Volts=$V_{m2}-(V_3$ or $V_4)$

Total length to be covered=(Number of degrees/Volt)*radius

Total control element is MU=Total length to be covered* Number of Volts/Radius in consideration

Dimensional Analysis: degree*l*V/(V*l)=degrees

Hence control element,

$$MU=180/(V_{max}-V_{min})$$

4.8 FPGA implementation of motor control

Stage-1: Finding difference between two voltages

$$V_{m1}= V3-V4$$

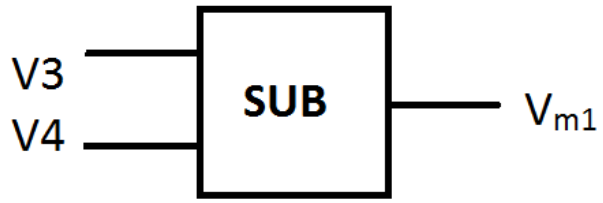


Fig 4.2 Stage-1 of motor control

Stage-2: Finding the average of two voltages

V_{m2}= (V3+V4)/2=(V3+V4)*0.5

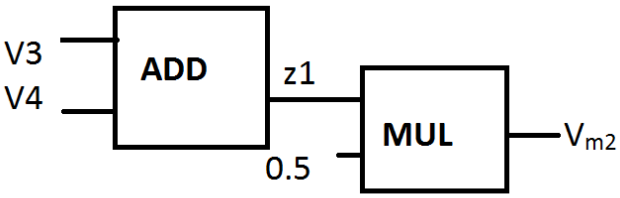


Fig 4.3 Stage-2 of motor control

Stage-3: Finding U

$U=V_{m2}-(V_3-V_4)$

V3 or V4 is selected by using a 2:1 Multiplexer with $V_{m1}(31)$ as select line. When V3<V4, then $V_{m1}(31)=1$ and V3 will be selected. When V3>V4, then $V_{m1}(31)=0$ and V4 will be selected.

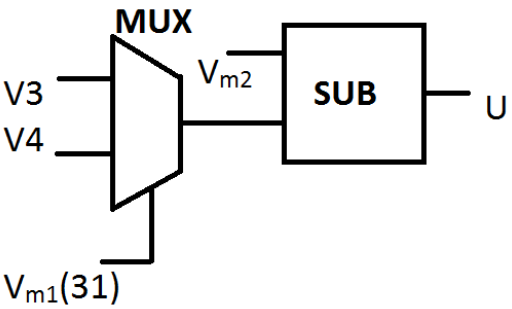


Fig 4.4 Stage-3 of motor control

Stage-4: Rotation priori estimate at t

$\Theta_{mt}^{-} = \Theta_{mt-1} \pm MU$

Add_sub block is used and its select line is again fed by $V_{m1}(31)$. When V3<V4, then $V_{m1}(31)=1$ and V3 will be selected. When V3>V4, then $V_{m1}(31)=0$ and V4 will be selected. Since our requirement is opposite of this, hence $V_{m1}(31)$ is negated before feeding it to the add_sub block as select line.

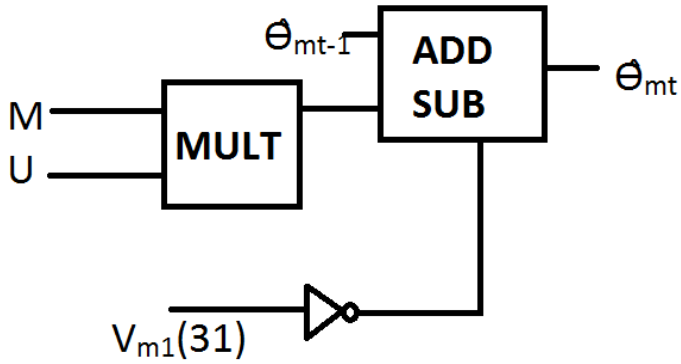


Fig 4.5 Stage-4 of motor control

Stage-5: Error covariance of priori estimate at t

$Z_t^{-} = Z_{t-1} + Q$

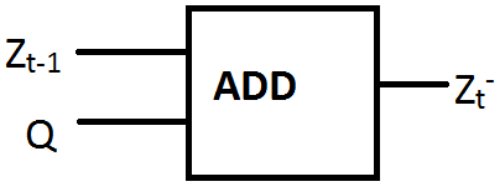


Fig 4.6 Stage-5 of motor control

Stage-6: Kalman Gain at t

$K_t = z_t^{-}(z_t^{-} + R)^{-1}$

$(z_t^{-} + R)$ is found and its inverse is found because division is more complex.

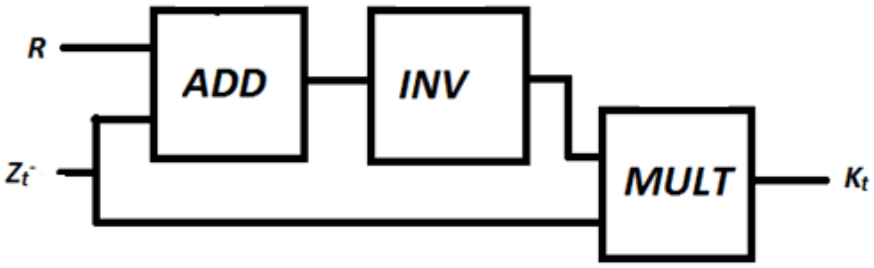


Fig 4.7 Stage-6 of motor control

Stage-7: Rotation posteriori estimate

$$\Theta_{mt}^- = \Theta_{mt} \pm K_t[\Theta_{reft} - \Theta_{mt}^-]$$

Add_sub block is used and its select line is again fed by $V_{m1}(31)$. When V3<V4, then $V_{m1}(31)=1$ and V3 will be selected. When V3>V4, then $V_{m1}(31)=0$ and V4 will be selected. Since our requirement is opposite of this, hence $V_{m1}(31)$ is negated before feeding it to the add_sub block as select line.

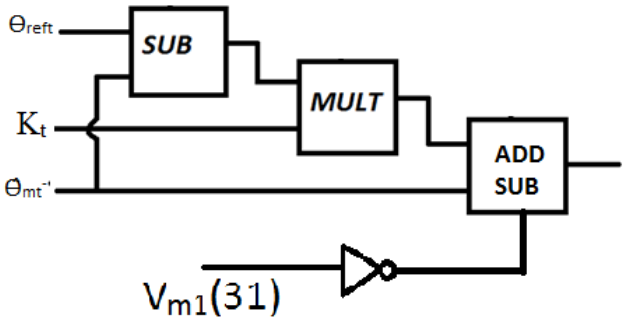


Fig 4.8 Stage-7 of motor control

Stage-8: Posteriori error covariance estimare

$$z_t = z_t^- - K_t z_t^-$$

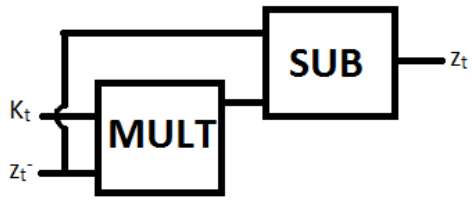


Fig 4.9 Stage-8 of motor control

Block diagram of motor control

To design entire block model using structural model

Equipment: Modelsim 6.6d altera starter edition and all components from Stage-1 through Stage-8

Input: clock=1 bit

V3=32bits

V4=32bits

Θ_{mt-1}=32bits

z_{t-1}=32bits

Output: Θ_{mt}=32bits

z_t=32bits

Intermediate signals required: I1-I14

Intermediate signals considered: I1, I3, I5, I7, I8, I9, I12

Intermediate constant signals: M, Q, R

*all intermediate signals are 32 bit wide

Signal mapping of intermediate signals

I1=V_{m1}

I3=V_{m2}

I5=U

I7= Θ_{mt}

I8= Θ_{reft}- Θ_{mt}

I9= z_t

I12=K_t

Other signals are used internally.

4.6 Cylinder algorithm

This is phase-2 of movement control. LDR sensors 1 and 2 are used as primary control inputs to the system. Voltages at hand are V_1 and V_2. To find necessary elevation, first we need to find out which voltage is more. The movement takes place towards the highest voltage side. The magnitude of translation is known from the difference of average of two known voltages and the difference of two known voltages. Ideally, the translation stops when the average value is equal to the lowest value.

Let $V_{t1} = V_1 – V_2$

$V_{t2} =(V_1 + V_2)/2$

When $V_{t1} >0$, V_1 is greater than V_2 translation takes place towards LDR sensor 1 and it stops when $V_2 = V_{t2}$

When $V_{t1} <0$, V_1 is less than V_2 translation takes place towards LDR sensor 2 and it stops when $V_1 = V_{t2}$

4.7 Kalman Filter for cylinder algorithm

Let L be the process and L_{t-1} be the known state measured variable is the high irradiance position at a particular time during the day.

Let it be L_{ref}

We have,

$$L_t = L_{t-1} + MU$$

Here, A=I and B=M

Therefore, translation estimate priori at t is

$$L_t = L_{t-1} + MU \qquad (1)$$

The process noise is assumed to be zero initially.

Prior estimate of error covariance is

$$Z_t = Z_{Zt-1} + Q \qquad (2)$$

Measured translation is

$$L_{ref\,t} = L_t + V \qquad (3)$$

Here C=I

Equation (1) and (2) form the prediction state of the control estimates are estimated now.

The kalman gain is calculated from the formula

$$K_t = K_t = z_t^- c^T / (c z_t^- c^T + R)$$

Since c=I in our requirement

$$K_t = z_t^- / (z_t^- + R)$$

$$K_t = z_t^- (z_t^- + R)^{-1} \qquad (4)$$

Posteriori translation estimate is

$$\hat{L}_{t=}\ \hat{L}_t^- + K_t[L_{reft} - \hat{L}_t^-] \qquad (5)$$

Posteriori error covariance estimate is

$$z_t = z_t^- - K_t z_t^-$$

$$z_t = z_t^- (1 - K_t) \qquad (6)$$

Equations (4), (5) and (6) for the correction step for the estimated estimates previously.

Control element:

From equation (1) in the previous algorithm,

$$\hat{L}_t = \hat{L}_{t-1} + MU$$

MU is a dimensional quantity with unity length.

Sources at hand are two voltages V_1 and V_2 and the radius of the panel in consideration. Assuming cylinder head contacts the panel at r/2 when r is the radius.

Control factor U = $V_{t2} - (V_1 \text{ or } V_2)$

The unit of U is volts thus V. Unit of M should be L/V

Ideal case: let V_1 has maximum and V_2 has minimum measurable voltage at panel

$(V_{max} - V_{min})$ V =180

1 V= 180/ $(V_{max} - V_{min})$ degrees

1V= 180/ $(V_{max} - V_{min})$ * (π/180)

1 V = π/$(V_{max} - V_{min})$ radians

Number of radians per volt = π/$(V_{max} - V_{min})$ radians

Number of volts = $(V_{t2} - (V_1 \text{ or } V_2))$ volts

Total length to be covered = (number of radians per volt) * (radius)

= π/$(V_{max} - V_{min})$ * r

Total control element is MU

= (total length to be covered/ radius) * (distance from centre to cylinder head) * (number of volts)

(1/r) * π/$(V_{max} - V_{min})$ * r * (r/2) * $(V_{t2} - (V_1 \text{ or } V_2))$

MU= r$[V_{t2} - (V_1 \text{ or } V_2)]$ / 2$(V_{max} - V_{min})$ * π

Therefore M= r / 2(V_{max} – V_{min}) * π

Where r is the radius of the panel

4.5 FPGA implementation of motor control

Stage-1: Finding difference between two voltages

V_{t1}= V1-V2

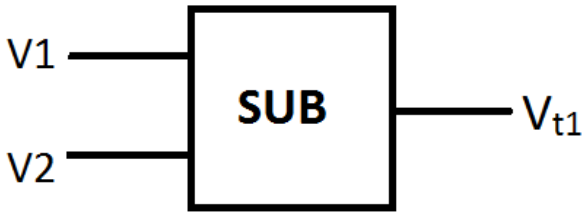


Fig 4.10 Stage-1 of cylinder control

Stage-2: Finding the average of two voltages

V_{t2}= (V1+V2)/2=(V1+V2)*0.5

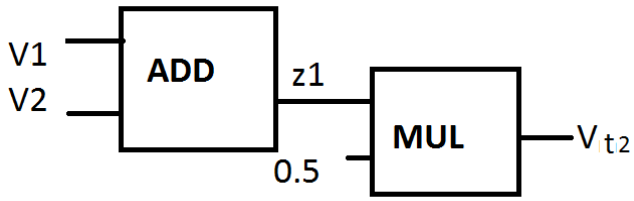


Fig 4.11 Stage-2 of cylinder control

Stage-3: Finding U

U=V_{t2}-(V_1-V_2)

V1 or V2 is selected by using a 2:1 Multiplexer with V_{t1}(31) as select line. When V1<V2, then V_{t1}(31)=1 and V2 will be selected. When V1>V2, then V_{t1}(31)=0 and V1 will be selected.

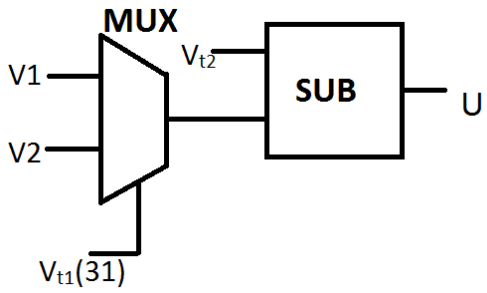


Fig 4.12 Stage-3 of cylinder control

Stage-4: Rotation priori estimate at t

$$\hat{L}_{pt}=\hat{L}_{pt-1}\pm MU$$

Add_sub block is used and its select line is again fed by $V_{t1}(31)$. When V1<V2, then $V_{t1}(31)=1$ and V1 will be selected. When V1>V2, then $V_{t1}(31)=0$ and V2 will be selected. Since our requirement is opposite of this, hence $V_{t1}(31)$ is negated before feeding it to the add_sub block as select line.

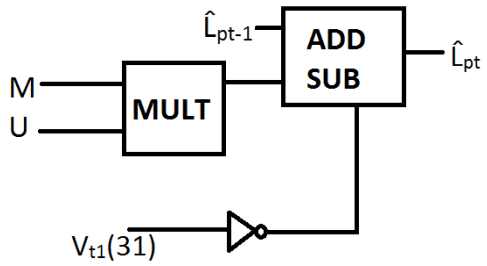


Fig 4.13 Stage-4 of cylinder control

Stage-5: Error covariance of priori estimate at t

$$Z_t^-=Z_{t-1}+Q$$

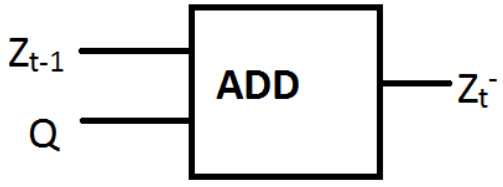


Fig 4.14 Stage-5 of cylinder control

Stage-6: Kalman Gain at t

$$K_t=z_t^-(z_t^-+R)^{-1}$$

(z_t^-+R) is found and its inverse is found because division is more complex.

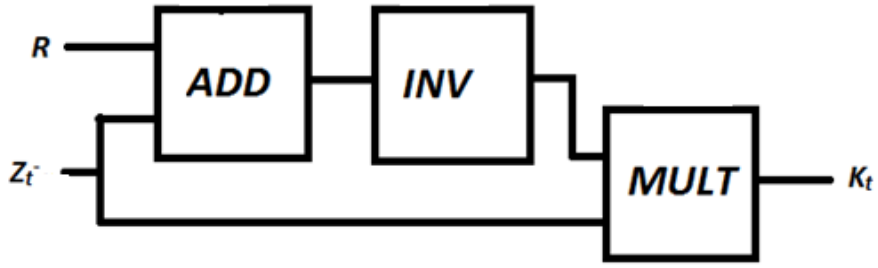


Fig 4.15 Stage-6 of cylinder control

Stage-7: Rotation posteriori estimate

$$\hat{L}_{t=}\ \hat{L}_t^-+K_t[L_{reft}-\hat{L}_t^-]$$

Add_sub block is used and its select line is again fed by $V_{t1}(31)$. When V1<V2, then $V_{t1}(31)=1$ and V1 will be selected. When V1>V2, then $V_{t1}(31)=0$ and V2 will be selected. Since our requirement is opposite of this, hence $V_{t1}(31)$ is negated before feeding it to the add_sub block as select line.

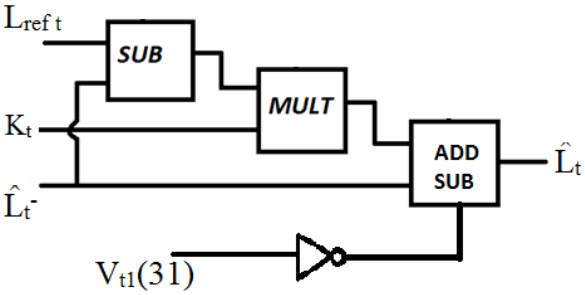


Fig 4.16 Stage-7 of cylinder control

Stage-8: Posteriori error covariance estimare

$$z_t=z_t^--K_tz_t^-$$

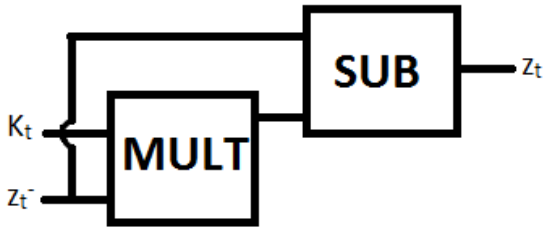


Fig 4.17 Stage-8 of cylinder control

Block diagram of cylinder control

To design entire block model using structural model

Equipment: Modelsim 6.6d altera starter edition and all components from Stage-1 through Stage-8

Input: clock=1 bit

V3=32bits

V4=32bits

$\hat{L}_{pt-1}$=32bits

z_{t-1}=32bits

Output: $\hat{L}_{pt}$=32bits

z_t=32bits

Intermediate signals required: I1-I14

Intermediate signals considered: I1, I3, I5, I7, I8, I9, I12

Intermediate constant signals: M, Q, R

*all intermediate signals are 32 bit wide

Signal mapping of intermediate signals

I1=V_{t1}

I3=V_{t2}

I5=U

I7= $\hat{L}_{pt}$

I8= L_{reft}- $\hat{L}_{pt}$

I9= z_t

I12=K_t

Other signals are used internally.

Chapter -5

FINITE STATE MODEL FOR SYSTEM

5.1 Finite State Machine Diagram

5.2 FSM Logic

5.3 State Encoding

5.1 Finite State Machine Diagram

The system is carried out in 5 states:

1. Initial: Here all the in-built values are taken but no output is taken. It forms the base step and keeps ready the data required for computation.
2. State1: This state carries out only the MPPT process. This state is used when there is no need for movement of motor and piston.
3. State2: This state carries out all the instructions in MPPT process and motor process. This is used when motor movement is required but not piston movement.
4. State3: This state carries out all the instructions in MPPT process and piston process. This is used when piston movement is required but not motor movement.
5. State4: This state carries out all the instructions in MPPT process, motor process and piston process. This is used when both motor and piston movement required.

One specific requirement is that MPPT process must run all along the time while motor and piston movement need not run all the time.

Initial stage is fed to the system reset. When reset is high, the system stops working and 0 is passed to all outputs.

5.2 FSM Logic

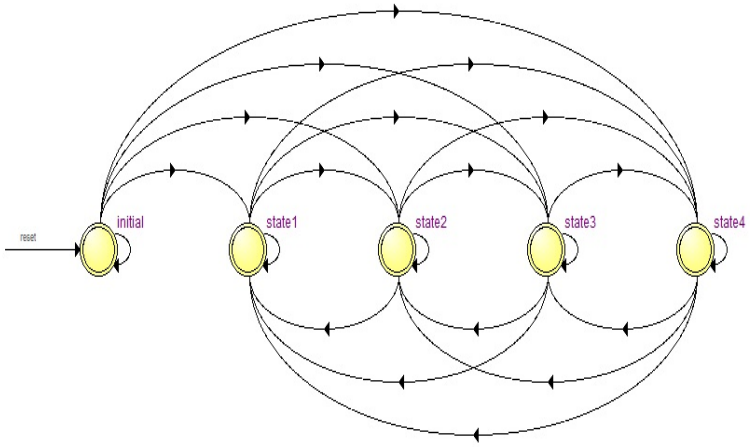


Fig 5.1 FSM Diagram

5.3 State Encoding

Name	State4	State3	State2	State1	Initial
Initial	0	0	0	0	0
State1	0	0	0	1	1
State2	0	0	1	0	1
State3	0	1	0	0	1
State4	1	0	0	0	1

Table 5.1 State encoding

Chapter-6

OBSERVATION TABLES AND SIMULATION

6.1 Observation Tables

6.2 RTL Schematics

6.1 Observation Tables

Table 6.1 Observations under partially shaded condition in fixed position ($+48^{o}$,$+30^{o}$)

Sl. No	Current (mA)	Voltage(v)	Time	Power (mA)
1	0.1	0.4	6:30	0.04
2	0.72	1.2	7:00	0.854
3	0.91	1.65	7:30	1.502
4	1.02	2.1	8:00	2.142
5	1.7	2.43	8:30	4.131
6	1.81	2.63	9:00	4.78
7	1.72	2.64	9:30	4.558
8	1.76	2.65	10:00	4.646
9	1.82	2.64	10:30	4.805
10	1.62	2.64	11:00	4.277
11	1.59	2.64	11:30	4.198
12	1.49	2.64	12:00	3.949
13	1.57	2.65	12:30	4.161
14	1.61	2.52	13:00	4.057
15	1.58	2.46	13:30	4.045
16	1.52	2.46	14:00	3.737
17	1.6	2.43	14:30	3.888
18	1.32	2.42	15:00	3.194
19	1.29	2.23	15:30	2.877
20	1.24	2.15	16:00	2.666
21	0.37	1.47	16:30	0.544
22	0.31	0.72	17:00	0.223
23	0.02	0.33	17:30	0.007
24	0	0	18:00	0

Table 6.2 Observations under partially shaded condition using algorithm (implementing MPPT)

Sl.No.	V_{ref} (V)	V_{act} (V)	V_{act} (V)	$V_m(1)$ (V)	$V_t(1)$ (V)	Current(mA)	Time	Position(in degrees)	Power(mVA) (V_{act} * current)
1	1.65	1.85	1.72	0.01	0.01	0.31	6:30	0,30	0.533
2	1.78	1.72.	1.87	0.15	0.13	0.82	7:00	0,30	1.533
3	2.3	1.87	2.32	0.08	0.03	1.01	7:30	5,35	2.343
4	2.52	2.32	2.68	0.07	0.02	1.12	8:00	12,35	3.002
5	2.68	2.68	2.71	0.01	0.18	1.9	8:30	18,35	5.149
6	2.69	2.71	2.72	0.11	0.07	1.83	9:00	23,38	4.978
7	2.69	2.72	2.72	0.02	0.02	1.71	9:30	30,40	4.651
8	2.69	2.72	2.72	0.13	0.02	1.80	10:00	38,45	4.896
9	2.68	2.72	2.72	0.07	0.01	1.73	10:30	45,45	4.706
10	2.69	2.72	2.72	0.07	0.005	1.41	11:00	62,45	3.835
11	2.69	2.72	2.72	0.13	0.13	1.62	11:30	75,48	4.406
12	2.69	2.72	2.72	0.18	0.11	1.74	12:00	85,48	4.733
13	2.69	2.72	2.72	0.01	0.07	1.71	12:30	90,48	4.651
14	2.69	2.72	2.72	0.01	0.08	1.69	13:00	95,48	4.597
15	2.69	2.72	2.72	0.08	0.04	1.71	13:30	103,48	4.651
16	2.69	2.72	2.72	0.19	0.03	1.53	14:00	110,45	4.162
17	2.67	2.72	2.72	0.20	0.19	1.48	14:30	118,45	4.025
18	2.68	2.72	2.72	0.02	0.11	1.63	15:00	125,45	4.4064
19	2.68	2.72	2.72	0.07	0.14	1.57	15:30	135,35	4.2704
20	2.57	2.72	2.70	0.11	0.17	1.28	16:00	140,35	3.456
21	2.52	2.70	2.66	0.10	0.08	1.3	16:30	150,35	3.497
22	2.49	2.69	2.68	0.02	0.01	1.09	17:00	150,30	2.921
23	1.82	2.68	2.41	0.01	0.01	0.41	17:30	150,30	0.988
24	0.87	2.41	1.93	0.01	0.01	0.23	18:00	150,30	0.444

6.2 RTL Schematics

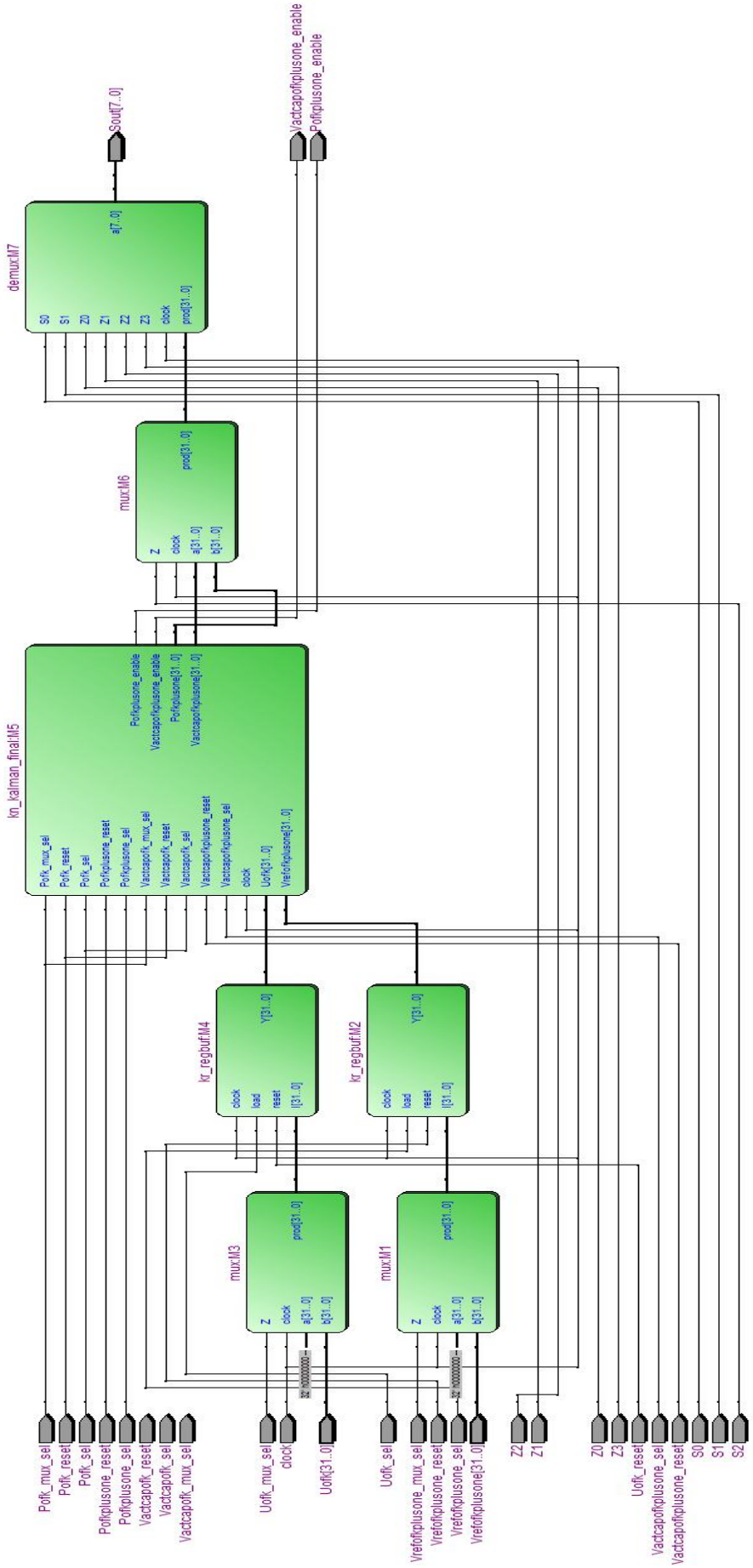


Fig 6.1 RTL Schematics for MPPT

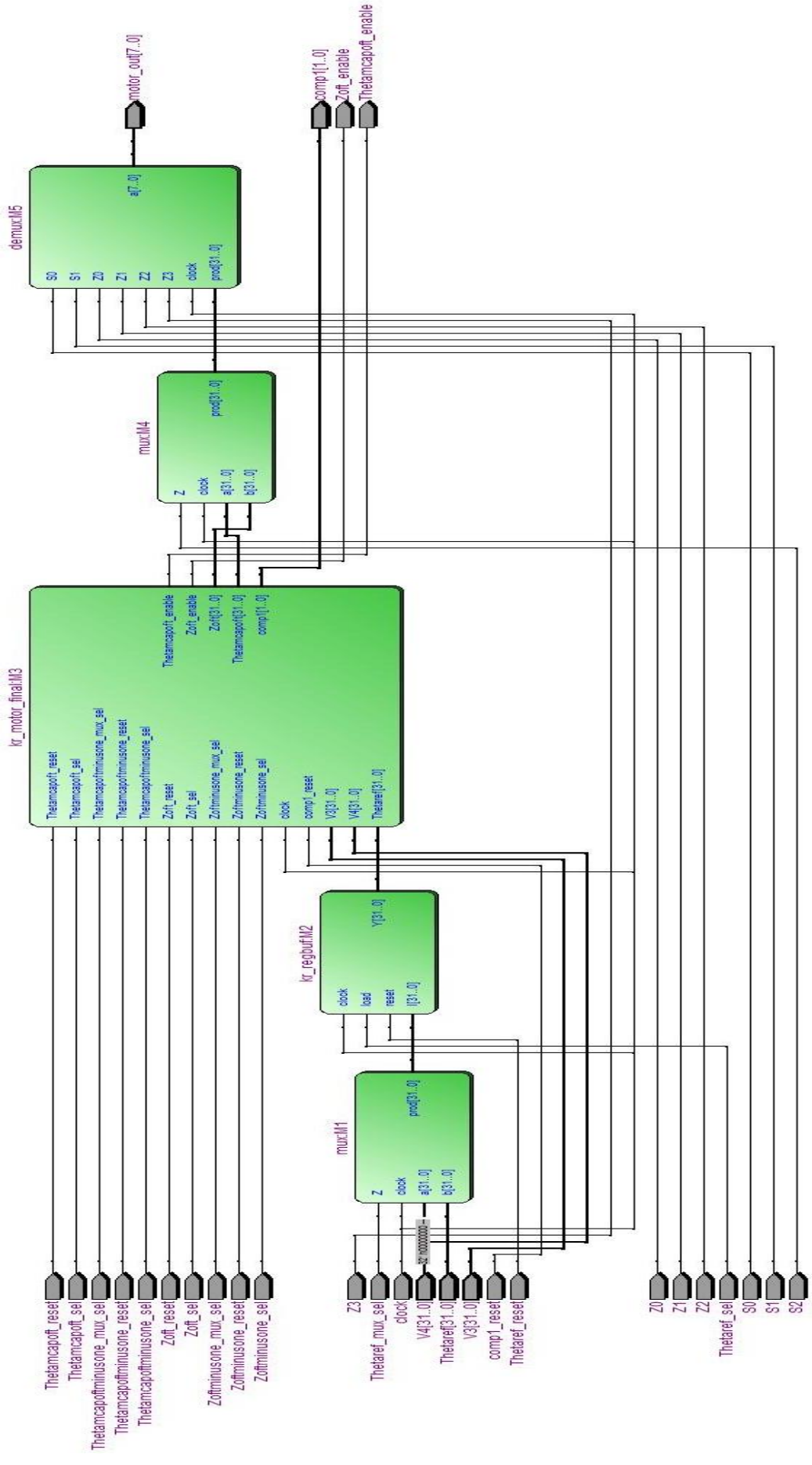


Fig 6.2 RTL Schematics for motor implementation

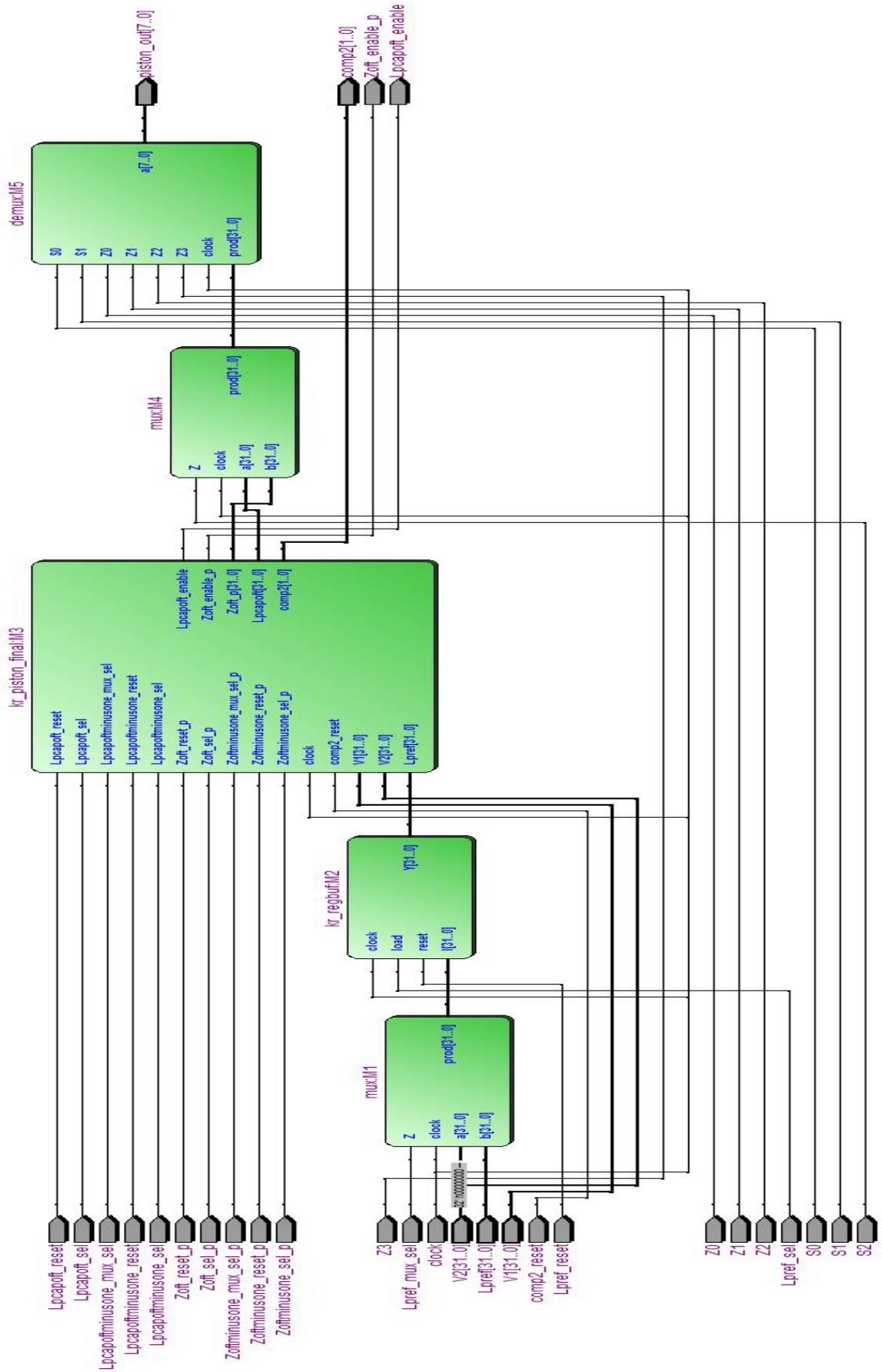


Fig 6.3 RTL Schematics or piston implementation

DISCUSSION

For implementation purpose, a 2.75V (open circuit voltage) and 2 mA (short circuit voltage) solar panel is used. It produces 5.5mW at 25^0c and 1KW/m^2 irradiance. MPP varies from 1.7V to 2.75V depending upon environmental conditions. MPPT algorithm, motor algorithm and piston algorithm are implemented individually on cyclone-II, EP2C20F484C7 as implementation in reconfigurable architecture like FPGA ensured hardware based flexibility. However, the computational complexity of all the three processes combined together and also the pin count is more than the capacity of EP2C20F484C7. So the total system has been simulated using cyclone-IV GX EP4CGXII0DF31C8.

Resource utilization:

Total logic elements- 29,794 (27%)

Total register - 21,571

Total pins - 304

Total memory bits - 6,954(<1%)

Embedded multiplier 9 bit elements – 164(29%)

Successful implementation fit this method coil eliminates the need of dedicated GPS hardware to the system to track the position of sun. This will be cost effective as no further internet packs are required. The only maintenance cost will be the war and tear of the motor joint which is also present in GPS based systems available in the market.

CONCLUSION

In this project, maximum power point tracking algorithm using Kalman filter has been proposed. Also, a dedicated algorithm for motor control tracking solar position using LDR sensors has been proposed including a dedicated algorithm for piston control tracking solar position using LDR sensors both based on Kalman filter. The maximum powerpoint has been tracked with an efficiency of 93.62% and the combined system of motor and piston algorithm tracked solar position with an error of 2%. Further work can be done by interfacing motor and piston using the code written.

REFERENCES

1. Shubhajit Roy Chowdhury, Dipankar Mukherjee, Hiranmay Saha. “FPGA based maximum power-point tracker of partially shaded solar photovoltaic arrays using modified adaptive perceptive particle swarm optimization”. International Journal on Smart Sensing and Intelligent Systems, vol. 2, no. 4. December 2009.
2. R.Kalman. “A new approach to linear filtering and prediction problems”. Journal of Basic Engineering, vol. 82, 1960.
3. Greg Welch and Gary Bishop. “An Introduction to the Kalman Filter”. University of North Carolina at Chapel Hill, Chapel Hill, NC 2006.
4. Kranthi Pamarthi, Varun Ramchandani. “Comparative study of MPPT using linear Kalman Filter for solar PV array on FPGA”. International journal on smart sensing and intelligent systems, 2012.
5. Kranthi Pamarthi, Varun Ramchandani. “Implementation of maximum power point tracking by using Kalman filter for solar PV array on FPGA”. IJSGCE, 2012.
6. Trishan Esram and Patrick L. Chapman. “Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques”. IEEE Transactions On Energy Conversion, Vol. 22, No. 2, June 2007.
7. Roberto Faranda and Sonia Leva. “Comparision of MPPT techniques for PV systems”. WSEAS Transcations on Power Systems.
8. B.O. Kang, J.H. Park. “Kalman filter MPPT method for a solar inverter”. Power and Energy Conference at Illinois (PECI), 2011 IEEE.

9. A. Mellit, H. Rezzouk, A. Messai, B. Medjahed. “FPGA-based real time implementation of MPPT-controller for photovoltaic systems”. Renewable Energy, Volume 36, issue 5 (May, 2011)